G-PARKING FUNCTIONS, ACYCLIC ORIENTATIONS AND SPANNING TREES

BRIAN BENSON, DEEPARNAB CHAKRABARTY, AND PRASAD TETALI

ABSTRACT. Given an undirected graph $G = (V, E)$, and a designated vertex $q \in V$, the notion of a G-parking function (with respect to q) was independently developed and studied by various authors, and has recently gained renewed attention. This notion generalizes the classical notion of a parking function associated with the complete graph. In this work, we study properties of *maximum G-parking functions* and provide a simpler new bijection between them and the set of spanning trees of G with no broken circuit. As a case study, we specialize some of our results to the graph corresponding to the discrete n-cube Q_n. We present the article in an expository self-contained form, since we found the combinatorial aspects of G-parking functions somewhat scattered in the literature, typically treated in conjunction with sandpile models and closely related chip-firing games.

1. INTRODUCTION

The classical parking functions provide a bijective correspondence between the spanning trees of the complete graph K_n and certain integer-valued functions on the vertices of K_n. A notion of parking functions corresponding to the spanning trees of an arbitrary graph G is more recent and has been independently developed in physics and combinatorics. It was introduced by Bak, Tang and Wisenfeld [1] as a self-organized sandpile model on grids, and was generalized to arbitrary graphs by D. Dhar [15]. See Definition 2.1 below for the precise definition of a G-parking function, associated with a connected graph G.

This notion is already rather powerful; besides generalizing the classical parking function from K_n to an arbitrary graph, it has been investigated in the context of chip-firing games [7, 26, 27] and the Tutte polynomial [8, 12] in discrete mathematics, and also investigated in algebra and related fields [2, 14, 28]. However, some of the combinatorial aspects of this topic appear somewhat scattered in the literature and are far from transparent. Several fundamental results concerning the recurrent configurations of chip-firing can be derived without the chip-firing context and terminology. For this reason, we shy away from introducing and discussing the chip-firing terminology. Instead, in this article we describe various interpretations of the G-parking functions in the most elementary combinatorial ways. Using a natural partial order $\prec$ on the set $\mathcal{P}(G, q)$ of parking functions, we consider the maximal elements in this poset $(\mathcal{P}(G, q), \prec)$. Much of our focus in this paper is on understanding the properties of such

maximal parking functions. The first result we describe (see Theorem 4.1) provides a simple bijection between the maximal parking functions in the poset and the set $\mathcal{A}(G; q)$ of acyclic orientations of G with a unique source at q. En route, we describe what we call an Extended Dhar algorithm (since it is an extension of an algorithm due to Dhar [15] to recognize G-parking functions) in providing an acyclic orientation corresponding to a maximal parking function. We review various combinatorial consequences and algebraic connections of this correspondence. For example, using known results (namely those of Greene and Zaslavsky [22] and more recent work of Gebhard and Sagan [19]), we further identify a 1-1 correspondence between the set of maximal parking functions and the set of spanning trees with no "broken circuits," or equivalently, the set of "safe" spanning trees ; see Section 4 for the definitions of these terms. In this paper, we provide a much simpler bijection (compared to [19]) between the set of safe trees and the set of acyclic orientations with a unique sink (or equivalently, a unique source). Furthermore we generalize this bijection to one between all spanning trees and all G-parking functions which *preserves* the bijection between safe trees and maximal G-parking functions. We must remark here that other bijective proofs between the set of G-parking functions and the set of spanning trees of G (for arbitrary connected G) have been given by Chebikin and Pylyavskyy [11]. However, to our knowledge, the simpler bijection we report here, in Theorem 4.2 below, and its generalization given in Theorem 4.6, are indeed new.

As an additional contribution, we describe a simple way to generate maximal parking functions in the Cartesian product graph $G_1 \square G_2$, using maximal functions in the (factor) graphs G_1 and G_2. We then specialize our study to understanding the parking functions in the discrete n-cube Q_n on 2^n vertices. By describing certain special constructions of maximal parking functions f on Q_n, we obtain a natural description of a set, $\text{dom}(f)$, of parking functions – those dominated, in the partial order given by $\prec$, by a special maximal parking function f. Interestingly enough we shall deduce (see Theorem 5.2) that

$$(1.1) \qquad |\text{dom}(f)| = \prod_{k=2}^{n} k^{\binom{n}{k}},$$

while it is a well-known fact that

$$(1.2) \qquad |\mathcal{P}(Q_n, q)| = \prod_{k=2}^{n} (2k)^{\binom{n}{k}} = 2^{2^n - n - 1} \prod_{k=2}^{n} k^{\binom{n}{k}}.$$

Recall that (1.2) corresponds to the total number of spanning trees of Q_n (see equation (5.85) in [29]), using the matrix-tree theorem and the explicit knowledge of the corresponding eigenvalues, to help evaluate the determinantal formula. In light of the fact that finding a bijective

proof accounting for the number of spanning trees of Q_n has been open for several years, we hope this is a nontrivial step towards such a proof.

The paper is organized as follows. In Section 2, we review some preliminaries, including Dhar's burn criterion, which determines whether a given function is a parking function. In Section 3, we show the bijection between maximum parking functions and acyclic orientations with a unique source. In Section 4.1, we describe our new and simpler bijection between the set of acyclic orientations with a unique sink and the set of safe trees. In Section 5.1, we describe a construction of maximum parking functions on Cartesian products of graphs. In Section 5.2, we focus our study on the *n*-cube Q_n, and provide some explicit constructions of maximum parking functions and related bounds. In Section 6, we provide the bijection between diffuse states and acyclic orientations of a graph. We conclude with some remarks on research in future directions and a few open problems in Section 7.

2. *G*-Parking Functions and Dhar's Burn Criteria

In this section we recall the definition of a *G*-parking function and review Dhar's (burning) algorithm that can be used to determine whether an integer-valued function on the vertices of *G* is a *G*-parking function.

Definition 2.1. *For a connected graph G, a G-parking function relative to vertex $q \in G$ is a function $f : V(G) \rightarrow \mathbb{Z}_{\geq -1}$ such that $f(q) = -1$ and for every non-empty $A \subseteq V(G)\backslash\{q\}$, there exists $v \in A$ such that $0 \leq f(v) < d_{\overline{A}}(v)$, where $d_{\overline{A}}(v)$ is the number of edges $e = vw$ with $w \notin A$.*

Remark 2.1. *Herein, we have modified the standard definition of a G-parking function somewhat. The function f is now defined on all of V(G) instead of restricted to simply $V(G)\backslash\{q\}$ in order to improve the compatibility between G-parking functions and Cartesian product graphs such as Q_n; due to this change, for f to be a G-parking function, $f(q) = -1$ necessarily.*

Proposition 2.1. *If for a function $f : V(G)\backslash\{q\} \rightarrow \mathbb{Z}_{\geq 0}$, for every non-empty* connected *subgraph $A \subseteq G\backslash\{q\}$, there exists $v \in V(A)$ such that $f(v) < d_{\overline{A}}(v)$, then f is a G-parking function.*

Proof. Assume that, for all connected $A \subseteq G\backslash\{q\}$, that there exists $v \in V(A)$ such that $f(v) < d_{\overline{A}}(v)$. Proceeding by contradiction, suppose that there is some disconnected $B \subseteq G\backslash\{q\}$ such that $f(v) \geq d_{\overline{B}}(v)$ for every $v \in V(B)$. Consider then any connected component C of B. Since C is connected we have, by the hypothesis of the proposition, that $f(v) < d_{\overline{C}}(v)$, for some vertex v in C. Thus $d_{\overline{B}}(v) < d_{\overline{C}}(v)$, implying that there is a vertex u in $\overline{C}\backslash\overline{B}$ such that v and u are

connected by an edge in G; otherwise, either $f(v) \geq d_{\overline{C}}(v)$ or $f(v) < d_{\overline{B}}(v)$. This contradicts the choice of C. ∎

Throughout we assume that the reference vertex q is fixed, and we always consider parking functions with respect to this fixed vertex q, without necessarily bringing explicit reference to it.

A natural question to ask is whether a given integer-valued function on the vertices of G can easily be tested for being a G-parking function. In the context of the so-called sandpile models, Dhar [15] provided an algorithm, which can be interpreted as an efficient algorithm to test if a given function is a G-parking function. This was observed in [11], wherein the algorithm was reformulated as follows. Let $f : V \setminus \{q\} \rightarrow \mathbb{Z}^+$. We assume that $f(q) = -1$.

Step 1. Mark any unmarked vertex v which has more marked neighbors than $f(v)$.

Step 2. Repeat Step 1 until no more vertices can be marked.

Step 3. Declare f to be a G-parking function if and only if all the vertices have been marked.

We omit the proof of correctness of the algorithm (as an exercise), which follows in a fairly straightforward way from the definition of a parking function.

On the set $\mathcal{P}(G, q)$ of parking functions on G with respect to q, there is a natural partial order we may define:

Definition 2.2. *Given two parking functions $f, g \in \mathcal{P}(G, q)$, we say $g < f$, if $g(v) \leq f(v)$, for all $v \in V$. The maximal elements in this partial order will be referred to as* maximal *parking functions. Finally, a parking function with the largest sum is called a* maximum *parking function.*

For $f \in \mathcal{P}(G, q)$, let $\|f\| := \sum_v f(v)$, and $\mathrm{dom}(f) = \{g \in \mathcal{P}(G, q) : g < f\}$. Furthermore, for $f, g \in \mathcal{P}(G, q)$, let $f \wedge g$ be the function on vertices, defined as $f \wedge g(v) := \min\{f(v), g(v)\}$, for each $v \in V(G)$.

While the following propositions are perhaps folk-lore, the first part of Proposition 2.2 and Proposition 2.3 appear as Lemmas 7 and 5 in [8].

Proposition 2.2. *(a) Let $f \in \mathcal{P}(G, q)$, and suppose $g : V(G) \rightarrow \mathbb{Z}_{\geq -1}$ such that $g(q) = -1$ and $0 \leq g(v) \leq f(v)$, for $v \in G$ with $v \neq q$. Then $g \in \mathcal{P}(G, q)$.*

(b) If $f, g \in \mathcal{P}(G, q)$, then $f \wedge g \in \mathcal{P}(G, q)$ and $\mathrm{dom}(f \wedge g) = \mathrm{dom}(f) \cap \mathrm{dom}(g)$.

Proof. While Part (a) is clear from the definition of a *G*-parking function, it can also be seen using Dhar's algorithm.

Note that $f \wedge g(q) = -1$ and so $f \wedge g \in \mathcal{P}(G, q)$, by Part (a). Now $f \wedge g \prec f$ and $f \wedge g \prec g$, so $h \in \text{dom}(f \wedge g)$ implies that $h \in \text{dom}(f) \cap \text{dom}(g)$. Finally consider, $h \in \text{dom}(f) \cap \text{dom}(g)$. Then at each vertex $v \in V(G)$, $h(v) \leq f(v)$ and $h(v) \leq g(v)$, so $h(v) \leq \min\{f(v), g(v)\} = f \wedge g(v)$, proving that $h \in \text{dom}(f \wedge g)$.

∎

Proposition 2.3. *For every connected graph* $G = (V, E)$, *every* $f \in \mathcal{P}(G, q)$, *we have* $\|f\| \leq |E| - |V|$. *More over, the equality is always achieved.*

Proof. We may prove this by induction on the number $n \geq 1$ of vertices G. The base case consisting of $V = \{q\}$ and no edges is trivially true. For the induction case, let $n \geq 2$. Given $f \in \mathcal{P}(G, q)$, let v_n be the last vertex to be marked by the Dhar algorithm. Then consider $H := G \setminus \{v_n\}$, the graph obtained by removing v_n and its incident edges. H is connected, since Dhar successfully marks all vertices before v_n, and more over, the function f_H denoting, f restricted to H, is an H-parking function with respect to $q \in H$. Thus we may apply the induction hypothesis to H and f_H and complete the proof:

$$\|f\| = f(v_n) + \|f_H\| \leq \sum_{v \neq v_n} f(v) \leq d(v_n) - 1 + (|E| - d(v_n)) - (|V| - 1) \leq |E| - |V|,$$

where we also used the fact that $f(v) \leq d(v) - 1$, for every v and parking function f.

The proof also suggests that by assigning the maximum possible value, at each step in Dhar's marking algorithm, one easily obtains a (maximum) parking function which achieves the upper bound. ∎

Note that the quantity $\mathbf{g}(G) := |E| - |V| + 1$ is sometimes referred to as the *cyclomatic number* or the *Betty number* of the graph and due to our convention of assigning $f(q) := -1$, we have $\mathbf{g}(G) - 1$, as the bound in the above proposition.

Proposition 2.2 also gives us a simple (albeit not necessarily efficient), inclusion-exclusion method to relate the set of maximum parking functions with the set of all parking functions. But this has to wait until the next section, where we observe another basic fact concerning the maximum parking functions.

3. Maximal G-Parking Functions and Acyclic Orientations with a Unique Source

Given a graph G, the notion of an acyclic orientation of the edges of G is classical, with an extensive literature. The notion of an acyclic orientation with a *unique source* at a fixed vertex is less well-studied. Let $\mathrm{AO}(G)$ denote the set of acyclic orientations of the graph G and let $\mathcal{A}(G, q)$ be the set of acyclic orientations of G with a *unique source* at vertex q. Finally, let $\mathrm{MP}(G, q)$ denote the set of maximum G-parking functions.

Theorem 3.1. *There exists a bijection between $\mathcal{A}(G, q)$ and the set $\mathrm{MP}(G, q)$ of maximum G-parking functions.*

Proof. Given an acyclic orientation $O \in \mathcal{A}(G, q)$ with a unique source at q, define the function $f = f(O)$ on the vertices of G: let $f(v)$ be the indegree (in O) of v minus 1. We will show that this correspondence provides the necessary bijection.

In any orientation, the sum of the indegrees equals the number of edges. Hence $\sum_v f(v) = |E| - |V|$. By using Dhar's algorithm, we may show that f is in fact a parking function: starting with q, we may repeatedly mark and remove the current set of source(s) in the acyclic orientation of the remaining graph; since a vertex v with value $f(v)$ becomes a source only when all its $f(v) + 1$ in-neighbors have been marked and removed, the Dhar criterion is satisfied. Also observe that the procedure stops only after marking all the vertices, since every acyclic orientation has at least one source. Thus $f(O) \in \mathrm{MP}(G, q)$.

To see that $f(O_1) \neq f(O_2)$, whenever $O_1 \neq O_2$, simply recall that an acyclic orientation is uniquely determined by its outdegree sequence: starting with the sinks, orient all edges into the sinks, remove the sinks, and repeat the process by subtracting one from the outdegrees of the neighbors of the sinks.

The proof will be complete once we establish the onto property, that every maximum parking function can be obtained this way. Given a maximum parking function $f \in \mathrm{MP}(G, q)$, we will construct an orientation $O(f)$ using the following modification of Dhar's algorithm, and will show that $O(f) \in \mathcal{A}(G; q)$, thus essentially providing an inverse map to the above construction.

The Extended Dhar Algorithm.

Input: A maximal parking function $f \in \mathrm{MP}(G, q)$

Output: An acyclic orientation $O(f)$ with a unique source at q.

Step 1. Start with $v = q$. Orient all edges out of q.

Step 2. If there exists a vertex v which accrued indegree(v) equal to $f(v) + 1$, mark v and orient the remaining edges incident at v outward from v.

Step 3. Repeat Step 2, until all vertices are marked and all edges are oriented.

The correctness of the original Dhar algorithm guarantees that all vertices will eventually be marked – indeed, the indegree(v) equals the number of neighbors marked before v; thus all edges will be oriented, meaning that $O(f)$ is an orientation of the edges of G. Observe that the indegree of a vertex v equals $f(v) + 1$. Since q is unique with $f(q) := -1$, it must be that q is the unique source. It is also easy to see that $O(f)$ is acyclic – if there were to be a cycle, considering the first vertex in the cycle which was marked, we obtain a contradiction to the way the edges were oriented (in Step 2 above) from a marked vertex.   ∎

Remark 3.1. *Upon completion of this work, we discovered (thanks to Matt Baker), that Theorem 3.1 can also be derived using chip-firing games: As described in* [21]*, the notion of a so-called* diffuse state *(introduced by* [23] *and see Definition 6.1 below) helps relate chip-firing configurations to acyclic orientations. Also thanks to an anonymous referee of an earlier version of this work* [6]*, we learned that Theorem 3.1 is Lemma 10 (under the name of* allowable *orientations) of Biggs* [8]*, where it is mentioned that this in fact goes back to an even earlier result of Greene and Zaslavsky* [22]*.*

It is now easy to observe the following fact (which appears as Lemma 8 in [7]).

Corollary 3.2. *Every maximal parking function is a maximum parking function.*

Proof. This follows from the proof of correctness of the Extended Dhar algorithm described above – if f were maximal, but not maximum, then there must be a vertex in Dhar's marking whose indegree is at least $f(v) + 2$. But then we can increase $f(v)$ by one, and obtain a valid parking function, contradicting the maximality of f.   ∎

We now return to prove the simple result that was promised at the end of the previous section.

Corollary 3.3. *Let G be a finite graph with k maximum G-parking functions. Then there exist G-parking functions $f_1, \ldots, f_n$ such that the number of G-parking functions is*

$$\sum_{i=1}^{n} \pm |\mathrm{dom}(f_i)|,$$

where $n = 2^k - 1$ and the sign + or − is uniquely determined by Proposition 2.2.

Proof. Let $g_1, g_2, \ldots, g_k \in \mathcal{P}(G, q)$ be all the maximum G-parking functions. Trivially, every non-maximal parking function is dominated by some maximal parking function, and Corollary 3.2 lets us observe that,

$$\mathcal{P}(G, q) = \cup_{i=1}^{k} \text{dom}(g_i) \,.$$

Now we may simply use the inclusion-exclusion formula to count the size of the union of the above k sets:

$$
\begin{aligned}
|\mathcal{P}(G, q)| &= |\cup_{i=1}^{k} \text{dom}(g_i)| \\
&= \sum_{i} |\text{dom}(g_i)| - \sum_{1 \leq i < j \leq k} |\text{dom}(g_i) \cap \text{dom}(g_j)| + \cdots \\
&\quad + (-1)^k |\text{dom}(g_1) \cap \text{dom}(g_2) \cap \cdots \cap \text{dom}(g_k)| \\
&= \sum_{i} |\text{dom}(g_i)| - \sum_{1 \leq i < j \leq k} |\text{dom}(g_i \wedge g_j)| + \cdots + (-1)^k |\text{dom}(g_1 \wedge g_2 \wedge \cdots \wedge g_k)| \,,
\end{aligned}
$$

which, upon using Proposition 2.2, completes the proof. ∎

It would indeed be interesting to see if the above corollary can be used in making progress towards obtaining a bijective proof for the number of spanning trees of Q_n, the n-dimensional hypercube. In Section 5.2, we take a modest step towards it.

Theorem 3.4. *Let G be a simple, connected graph. Then, for a fixed choice of q, G has a unique maximum G-parking function if and only if G is a tree.*

Proof. If G is a tree, then there is only one parking function with respect to any q since G has no cycles and, thus, each vertex can have at most one marked neighbor in the Dhar algorithm. (Note that this is in fact tautological if one uses the bijection between the parking functions and the spanning trees of G.) Hence there is only one maximum G-parking function.

The other direction is less obvious. However, observe that in light of the bijection established in Theorem 3.1, it suffices to show the following. *Whenever G is connected and contains a cycle, then there are at least two acyclic orientations for G, with q as the unique source.* This is easy to establish (for example, by considering the standard directed acyclic graph (DAG) representation of the graph), and we leave the proof as a simple exercise. ∎

See Remark 4.1 below for another short (but indirect) proof of the above theorem.

Corollary 3.5. *A simple, connected graph G has a unique maximum parking function f if and only if the range of f is a subset of $\{-1, 0\}$.*

Proof. This follows from the observation that any tree has a unique parking function and, for any vertex $v \neq q$, $f(v) = 0$. ∎

Corollary 3.6. *For every G,* $|\mathcal{P}(G, q)| = 1$ *if and only if* $|\mathrm{MP}(G, q)| = 1$.

4. Bijections of Maximal *G*-parking Functions

We now augment our Theorem 3.1 above to show a bijection between maximal *G*-parking functions and spanning trees with no broken circuits. Before stating the theorem, we need to recall several definitions from the literature.

The Tutte (or Tutte-Whitney) polynomial of a graph $G = (V, E)$ is the two-variable polynomial defined as

$$(4.1) \qquad \mathcal{T}_G(x, y) = \sum_{A \subseteq E} (x - 1)^{\kappa(A) - \kappa(E)} (y - 1)^{|A| - n + \kappa(A)},$$

where $n = |V|$ and $\kappa(A)$ denotes the number of connected components of the graph on V using edgeset A. For $\lambda \in \mathbb{Z}^+$, the chromatic polynomial $\chi_G(\lambda)$ of a graph G is defined as the number of proper vertex colorings of G using λ colors. For a general variable λ, the following relation between the chromatic polynomial and the Tutte polynomial is well-known (see e.g. [4, 5, 30, 31, 32]):

$$(4.2) \qquad \chi_G(\lambda)) = (-1)^{n - \kappa(G)} \lambda^{\kappa(G)} \mathcal{T}_G(1 - \lambda, 0),$$

where $n = |V|$ is the number of vertices and $\kappa(G)$ is the number of connected components of G.

Definition 4.1. *Given a graph $G = (V, E)$ and an ordering of all the edges of G, a broken circuit $B \subseteq E$ is any cycle (of edges) of G minus the largest (according to the ordering) edge in the cycle.*

Note that since every cycle contains (or gives rise to) a broken circuit, a collection of edges *not* containing a broken circuit must necessarily be acyclic. Inspired by the terminology of Kenyon and Winkler [25], we call a spanning tree *T safe*, if no edge forms a broken circuit with *T*. The notion of a broken circuit is more general, and in fact explains the classical terminology: in the context of a matroid, an independent set of elements of the matroid obtained from a circuit, by removing the largest element (once again, according to some apriori global ordering of all the elements) of the circuit.

Theorem 4.1. *For every undirected, connected graph G, the following quantities are all the same.*

- *(a) The number of maximum G-parking functions with respect to q.*
- *(b) The number of acyclic orientations with a unique source at q.*
- *(c) The number of spanning trees with no broken circuits, or equivalently, with zero external activity.*
- *(d) The coefficient (up to sign) of the λ-term in the chromatic polynomial $\chi_G(\lambda)$.*
- *(e) The value (up to sign) $\mathcal{T}_G(1, 0)$ of the Tutte polynomial $\mathcal{T}_G(x, y)$, evaluated at $x = 1$ and $y = 0$.*

Proof. The equivalence between (c) and (d) is classical and is part of Whitney's Broken Circuit theorem [32]: that the chromatic polynomial on *n* vertices is given by

$$\chi_G(\lambda) = \sum_{r=0}^{n-1}(-1)^j m_r \lambda^{n-r},$$

where m_j is the number of *r*-subsets of edges of *G* which contain no broken circuit. The term m_{n-1} corresponds to (the absolute value of) the coefficient of λ; note that the $n-1$-subsets under consideration being necessarily acyclic, correspond to spanning trees which do not contain a broken circuit.

The equivalence between (b) and (d) is due to Greene and Zaslavsky [22]. A (direct) bijective proof of the equivalence between (b) and (c) is given by Gebhard and Sagan [19], using a modification of an algorithm of Blass and Sagan [10]. In Section 4.1 we provide a much shorter proof of the equivalence between (b) and (c).

The equivalence between (d) and (e) follows from (4.2), and using $\kappa(G) = 1$, for a connected *G*. The equivalence between (a) and (e) follows from results of [15] and [26], which confirmed a conjecture of Biggs [7] in the context of chip-firing. An inductive proof (using edge deletions and contractions) without involving chip-firing is due to Plautz and Calderer [27]. As described in [27], the work of Dhar and Lopez provides the following result:

$$\mathcal{T}_G(1, y) = \sum_{f \in \mathcal{P}(G, q)} y^{w(f)},$$

where $w(f) = |E(G)| - |V(G)| + \|f\|$, hence the equivalence of (a) and (e). The results in [26] and [12] also establish the equivalence between (c) and (e), with the minor modification that broken circuits are equivalently described using external acitivities - each broken circuit contributes an external activity of one to a spanning tree. ∎

In addition to the above, Cori and Le Borgne [12] describe certain decreasing traversals of vertices and edges and a notion of strong edges to provide a bijection between recurrent chip-firing configurations (with a fixed "level") and spanning trees with a fixed "external activity". While the level corresponds to the sum of the values of a parking function (up to an additive shift), the external activity reflects the number of broken circuits, and we refer the interested reader to their paper for additional information.

Remark 4.1. *Observe that the nontrivial part of Theorem 3.4 follows easily using the equivalence between (a) and (c). Indeed, let G be a connected graph which contains a cycle of length 3 or more. Then given a spanning tree which contains no broken circuit, we include an edge not in the tree to form a cycle C. Since C is of length 3 or more, there must be an edge, which is not the largest edge in the cycle, that can be removed, giving another spanning tree with no broken circuits. Hence a (connected) graph containing a cycle has more than one parking function.*

4.1. **Bijection between Acyclic Orientations with Unique Sink and Safe Trees.** In this section we give a shorter proof, of equivalence of (b) and (c) of Theorem 4.1, than the one reported in Gebhard and Sagan [19]. For ease of presentation, we will consider orientations with a unique sink rather than a unique source – clearly this is equivalent.

Let σ be any total ordering of the edges of G. Given two edges e and f, we say e is *larger* than f if $\sigma(e) > \sigma(f)$. Similarly we say e is *smaller* than f, if the inequality is otherwise. Recall, a spanning tree T of G is called *safe* with respect to σ if for any edge $e \notin T$, there exists at least one edge f in the unique cycle in $T + e$ such that f is larger than e. Let $\mathcal{T}(G, \sigma)$ be the set of safe trees with respect to σ, and let $\mathcal{A}(G, q)$ be the set of acyclic orientations of G with q being the *unique sink*.

Theorem 4.2. *For any total order σ, there exists a bijection $\mu_\sigma : \mathcal{T}(G, \sigma) \to \mathcal{A}(G, q)$.*

Henceforth, we fix σ and do not write it as a subscript. Before we proceed, we make a few more definitions and observations. An arborescence with root q is a directed spanning tree with all vertices except the root having out-degree exactly 1, and the root having out-degree 0. Any spanning tree corresponds to a unique arborescence with root q and henceforth we will use the terms interchangeably. Given a vertex i, we let P_i denote the unique directed path from i to q. Given vertex i and j, we let *meet*(i, j) be the first vertex in the intersection of P_i and P_j. That is, the path from i to *meet*(i, j) and the path from j to *meet*(i, j) are disjoint except at *meet*(i, j). If j lies on P_i, we let j be *meet*(i, j). Observe that for three vertices i, j, k, either

$meet(i, k) = meet(j, k)$, or $meet(i, k)$ lies on P_i and $meet(j, k) = meet(i, j)$, or $meet(j, k)$ lies on P_j and $meet(i, k) = meet(i, j)$. Given two vertices i and j, we will denote the largest edge in the path from i to $meet(i, j)$ as e_{ij} and the largest edge from j to $meet(i, j)$ as e_{ji}. If $j = meet(i, j)$, we let e_{ji} be the null edge.

We (abuse notation and) say $i >_\sigma j$ if e_{ij} is larger than e_{ji}. We will also define e_{ii} to be a null edge. Note that if $i >_\sigma j$ and $j >_\sigma k$, then going over the three possibilities of $meet(i, k)$ we see that $i >_\sigma k$. Thus $>_\sigma$ is transitive and induces a total ordering of vertices. We say i *dominates* j, if $i >_\sigma j$. It is instructive to note that i dominates all vertices in P_i.

We now describe a mapping μ from all arborescences with root q to acyclic orientations with unique sink q. We will prove that distinct safe trees lead to distinct arborescences. This proves that the mapping μ restricted to safe trees is one-to-one. Furthermore, given an acyclic orientation, we describe a procedure π which takes an acyclic orientation and returns a safe tree. Moreover, for any orientation $O \in \mathcal{A}(G, q)$, we have $\mu(\pi(O)) = O$. This shows that μ is onto and thus it is a bijection.

Arborescence to Acyclic Orientation (μ): Given an arborescence, orient an edge (i, j) as i to j if $i >_\sigma j$, or vice-versa. By the transitivity of $>_\sigma$ it is clear that the orientation is acyclic. Also every vertex dominates the root which therefore is the unique sink. We will call an orientation so obtained as one *induced* by the arborescence. The following lemma will show that two safe arborescences cannot lead to the same orientation.

Lemma 4.3. *Let T_1 and T_2 be two distinct safe arborescences and $O_1 = \mu(T_1)$ and $O_2 = \mu(T_2)$. Then $O_1 \neq O_2$.*

Proof. We prove the contrapositive: suppose $O_1 = O_2 = O$, then we show that $T_1 = T_2$. Consider the trees rooted at q (note that q is the unique sink of O) with edges directed towards q. We now show that for each vertex of the graph:

(*) The unique out-neighbor in T_1 is the same as that in T_2 which will imply that both trees are the same.

Since O is acyclic with a unique sink, the vertices V can be decomposed as $V = (q =: S_0 \cup S_1 \cup ... \cup S_r)$ for some $r \geq 1$, where S_i is the set of vertices which are sinks in the digraph $G \setminus \bigcup_{\ell=0}^{i-1} S_\ell$. Let S_i be the first set (with the least i) to contain a vertex violating (*). Let this vertex be denoted i (abusing notation). Let (i, j) and (i, k) be the unique out-neighbors of i in T_1 and T_2 respectively, with $j \neq k$, $j \in S_j$ and $k \in S_k$ (again abusing notation). Observe

that $j, k < i$ (that is, S_j, S_k precede S_i), by the nature of the decomposition, and the fact that $O = \mu(T_1)$ and $O = \mu(T_2)$.

Now consider the undirected cycle using the edges $(i, j), (i, k)$ and the unique paths, P_j from j to q and P_k from k to q. Observe that both P_j and P_k are contained in T_1 *and* T_2, by the choice of i. Also note that the largest edge in this cycle must be either on P_j or on P_k, since both T_1 and T_2 are safe! We now get a contradiction – if the largest edge is on P_j, then our definition of μ demands that (i, j) be oriented from j to i in T_1; similarly if it is on P_k, then the edge (i, k) be oriented from k to i in T_2. $\square$

Acyclic Orientations to Safe Trees(π): Now we describe a procedure to get a safe tree from an acyclic orientation O with single sink q. Let $d(v)$ denote the out-degree of vertex v. Note that $d(q) = 0$. For an edge oriented i to j, we say j is a out-neighbor of i and i is an in-neighbor of j. At each step we maintain a set of *labeled* vertices X and an arborescence T spanning X. We maintain the invariant that there is no edge from a vertex in X to a vertex in $Y := V \setminus X$, (think of X as a large sink). Initially, $X = \{q\}$. Note that since the orientation is acyclic, at any step there is at least one vertex u in Y which is a sink in the induced graph $G[Y]$, that is, u has no out-neighbors in Y. We pick one such u arbitrarily. Also, since there is a unique sink q, this vertex u must have at least one out-neighbor in X. We add u to X and we connect u to the arborescence T as follows.

Let $X_u \subseteq X$ be the set of out-neighbors of u. Let $x \in X_u$ be the vertex which dominates all other vertices in X_u with respect to the current arborescence T. Let $W_u \subseteq X_u$ be the subset of all vertices v such that $(u, v) >_\sigma e_{xv}$. That is, the edge (u, v) is larger than the largest edge in the path from x to *meet*(x, v). Note that $x \in W_u$ and thus W_u is non-empty. Connect u to the vertex v with the largest (u, v) among all v in W_u. We end when $X = V$ with an arborescence T.

Lemma 4.4. *The arborescence T obtained at the end is a safe tree. Moreover, $\mu(\pi(O)) = O$, for $O \in \mathcal{A}(G, q)$.*

Proof. Consider an edge (j, i) not in the tree. Note that at each step exactly one node is added to the arborescence. Also note that for an arc oriented (j, i) in O, i is added before j. Let X_j be the set of labeled vertices in the step when j is added to the arborescence. Note that $i \in X_j$. Suppose $x \in X_j$ was the dominator of X_j and (j, k) was the edge added at this step.

Observe that whenever the procedure adds a new vertex j, since it attaches itself to the largest vertex in $W_j \subseteq X_j$, it dominates all the other vertices in the arborescence. This is because $(j, k) >_\sigma e_{xk}$, that is, the largest edge in the path x to *meet*$(x, k) = $ *meet*(x, j) and thus

j dominates x and so every other vertex. Thus the tree T induces the same orientation O. It remains to show that T is safe, that is, (j, i) is not the largest edge in the cycle $T + (j, i)$.

If $i \in W_j$, then by choice of k, $(j, k) >_\sigma (j, i)$ and thus (j, i) is not the largest cycle in $T + (j, i)$.

If $i \notin W_j$, this means $(j, i) < e_{xi}$. That is, (j, i) is smaller than the largest edge in the path from x to $meet(x, i)$. Also, by definition, $(j, k) > e_{xj}$ that is, (j, k) is larger than the largest edge in the path from x to $meet(x, k)$. Let the path from x to $meet(x, i)$ be Q_i and that from x to $meet(x, k)$ be Q_k.

Now consider $meet(i, k)$. If $meet(i, k)$ does not lie on P_x, then $meet(x, i) = meet(x, k)$. Thus, $(j, k) >_\sigma (j, i)$. If $meet(i, k)$ lies on P_x, then it must be $meet(x, i)$ or $meet(x, k)$. If the latter, then $Q_i \subseteq Q_k$, and therefore $(j, k) >_\sigma (j, i)$. If $meet(i, k) = meet(x, i)$, then $Q_k \subseteq Q_i$. Either the largest edge in Q_i lies in Q_k and we are done as before; Or, the largest lies in $Q_i \setminus Q_k$, which lies in the cycle formed in $T + (j, i)$. Thus, (j, i) is smaller than the largest edge in the cycle of $T + (j, i)$, completing the proof.   □

4.2. **Extension to a bijection of G-Parking Functions and Spanning Trees.** In this section we generalize the above bijection between maximal parking functions (or equivalently acyclic orientations with unique sink) and safe trees to a bijection between G-parking functions and all spanning trees of G which preserves the bijection between maximal parking functions and safe trees. We use the definitions of the previous section and make a few more definitions and claim below before demonstrating the bijection.

Given an arborescence T spanning only a subset of vertices $X \subseteq V$, and a vertex $u \notin X$, we make a few definitions and observations which will be useful in our bijection. Let $\Gamma(u)$ be the neighbors of u and let $X_u := \Gamma(u) \cap X$. We now describe an order on the vertices X_u (which could be different from the total order of the previous paragraph, but is related), which we call the *power order* of X_u. Intuitively, given two neighbors v and w of u in X_u, if v is more powerful than w, then in the tree $T + (u, v)$, u would dominate w *and* the edge (u, w) doesn't form a broken circuit with $T + (u, v)$. Moreover, in the tree $T + (u, w)$, either v dominates u *or* the edge (u, v) forms a broken circuit with $T + (u, w)$.

Let x be the vertex in X_u which dominates all other vertices in X_u with respect to the current arborescence T. Let $W_u \subseteq X_u$ be the subset of all vertices v such that $(u, v) >_\sigma e_{xv}$. That is, the edge (u, v) is larger than the largest edge in the path from x to $meet(x, v)$. Note that $x \in W_u$ and thus W_u is non-empty. Let v be such that (u, v) is largest among all v in W_u. Call v the most

powerful element of X_u. Delete v from X_u and repeat till one gets an order on all vertices of X_u. We call this order the *power order w.r.t T*. In Figure 1 we give an illustrative example.
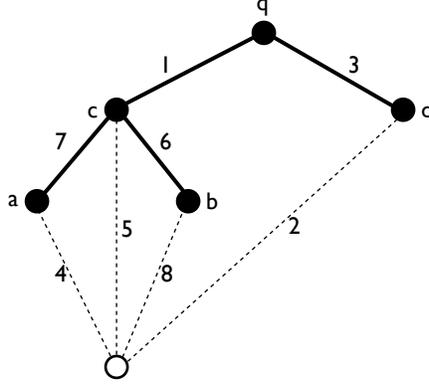
FIGURE 1. The numbers on the edges correspond to the σ value. The dark edges form the tree T and $u \notin T$. Note that $a >_\sigma b >_\sigma d >_\sigma c$ in T. However, the power order of $X_u = \{a, b, c, d\}$ is as follows. Firstly, $x = a$ and $W_u = \{a, b\}$. Since (u, b) is bigger, b is the most powerful. Second powerful is a. After that $x = d$ and $W_u = \{c, d\}$, and thus the complete power order is (b, a, c, d).

Claim 4.5. *Given a tree T and a vertex $u \notin T$ with neighbors X_u in T, and two vertices v and w in X_u. If v is more powerful than w, then u dominates w in the tree $T + (u, v)$ and (u, w) doesn't form a broken circuit with $T + (u, v)$. In the tree $T + (u, w)$, either v dominates u or (u, v) forms a broken circuit with $T + (u, w)$.*

Proof. Consider the time when the power order of v is determined. Let x be the dominator at that stage and let X_u the set of neighbors of u remaining. Note $w \in X_u$. Either $v \in W_u$ or $v = x$ and W_u is empty. If the former, (u, v) is larger than e_{xv} and thus in $T + (u, v)$, u dominates x which dominates w. Moreover (u, w) is either smaller than (u, v) if $w \in W_u$, or smaller than e_{xw} which is smaller than e_{uw} since u dominates x. In any case, (u, w) doesn't form a broken circuit in $T + (u, v)$. Also, in $T + (u, w)$, (u, v) forms a broken circuit since it is larger than the largest edge from v to w.

If the latter, then u dominates w in $T + (u, v)$ since $v = x$ dominates w in T. Also since W_u is empty, the edge (u, w) is smaller than e_{vw} implying (u, w) doesn't form a broken circuit in $T + (u, v)$. Also, in $T + (u, w)$, v dominates u since (u, w) is smaller than e_{vw}.   □

Now we are ready to present the next theorem which gives the desired bijection between G-parking functions and spanning trees of G. Let $\mathcal{T}(G)$ be the set of all spanning trees of G. Recall that $\mathcal{T}(G, \sigma)$ was the set of safe spanning trees with respect to σ.

Theorem 4.6. *There exists a bijection $\pi : \mathcal{P}(G, q) \rightarrow \mathcal{T}(G)$ such that for all $f \in \mathcal{P}(G, q)$ which is maximal, $\pi(f)$ is a safe tree.*

As in the proof of Theorem 4.1, we describe mappings $\mu : \mathcal{T}(G) \rightarrow \mathcal{P}(G, q)$ and $\pi : \mathcal{P}(G, q) \rightarrow \mathcal{T}(G)$ and show that for any parking function f, $\mu(\pi(f)) = f$ (Lemma 4.10) and for any spanning tree T, $\pi(\mu(T)) = T$ (Lemma 4.11). Furthermore we show that for a tree T, $\mu(T)$ is a maximal parking function iff T is safe (Proposition 4.7). This completes the proof of Theorem 4.6

Spanning Trees to Parking Functions (μ): Given a spanning tree T, let E' be the subset of edges not in T which form a broken circuit with T. Delete E' from E. Direct the edges of T with q as the unique source – that is, all vertices except q have an out-degree of 1 and q has an out-degree of 0. Given this arborescence, for any undirected edge (i, j) in $E \setminus E'$, orient it as (i, j) if $i >_\sigma j$, or (j, i) if $j >_\sigma i$. This gives di-graph D. The parking function $f := \mu(T)$ is defined as $f(v) = \text{out-degree}_D(v) - 1$ in the orientation of the edges of $E \setminus E'$.

Proposition 4.7. $f := \mu(T)$ *is a maximum parking function iff T is a safe tree.*

Proof. Note that if T is safe, E' is empty and therefore all the edges are oriented. Thus $\sum_v f(v) = |E| - |V|$ and by Proposition 2.3, f is a maximal parking function. On the other hand if T is not safe, then E' is not empty and thus $\sum_v f(v) < |E| - |V|$ implying f is not a maximal parking function. $\qquad\square$

Parking Functions to Spanning Trees (π): Given a valid parking function, we use a modification of Dhar's algorithm to obtain the spanning tree. We maintain a set of vertices X_i connected via an arborescence T_i, with X_0 initialized to $\{q\}$ and T_0 is $\varnothing$. At each step we add one vertex to X_i and one edge to T_i. In the end we get $X_{n-1} = V$ and T_{n-1} is the spanning tree returned. We describe the $i + 1$th step. Let $\Gamma(v)$ denote the neighbors of v.

(1) Let $S_{i+1} := \{v \in V \setminus X_i : |\Gamma(v) \cap X_i| > f(v)\}$.
(2) For every $u \in S_{i+1}$, $X_u := \Gamma(u) \cap X_i$ and let $M(u)$ be the $(|X_u| - f(u))$th vertex in the power order of X_u with respect to T. Let Y_u be the vertices in X_u more powerful than $M(u)$ and E_u be the set of $|X_u| - f(u) - 1$ edges of the form (u, v) where $v \in Y_u$.
(3) Note that adding all edges of the form $(u, M(u))$ to T_i gives a new tree T'. In T', let u be the vertex in S_{i+1} which is dominated by all other vertices in S_{i+1} with respect to T'. Add u to get X_{i+1} and the edge $(u, M(u))$ to get T_{i+1}.

Claim 4.8. *Given a parking function f, let the vertices be added in order $\{q = u_0, u_1, \cdots, u_{n-1}\}$. That is $X_i := \{u_0, \cdots, u_i\}$. Then with respect to the tree T_i, u_i dominates u_{i-1} which dominates u_{i-2} and so on.*

Proof. The proof is by induction. At stage i, let S_i be the set of vertices as defined above. Suppose u_i is added at this stage. We will be done if we show u_i dominates u_{i-1}.

Two cases arise: If u_{i-1} is not a neighbor of u_i, then u_i must have been in S_{i-1} as well implying u_i dominated u_{i-1} in T'_{i-1} and since $M(u_i)$ doesn't change, dominates u_{i-1} in T_i as well.

If u_{i-1} is a neighbor of u_i, then either $|X_{u_i}| = f(u_i) + 1$ and u_i connects to the most powerful of its neighbors in X_{u_i} and thus dominates u_{i-1} which is in X_{u_i}. Or, $|X_{u_i}| > f(u_i) + 1$, which once again implies u_i was in S_{i-1} and moreover, either $M(u_i)$ becomes u_{i-1}, or still remains more powerful than u_{i-1}. In any case, u_i dominates u_{i-1} in T_i.   □

Claim 4.9. *For any vertex u, the set of edges E_u are precisely the set of edges from u to X_u which form broken circuits with T.*

Proof. Consider vertex u connecting to vertex w in X_u. From the first part of Claim 4.5 we see that the edges not in E_u do not form broken circuits. Let (u, v) be an edge in E_u. Since v is more powerful than w, from the second part of Claim 4.5 we see either (u, v) forms a broken circuit with T or v dominates u in T. The second possibility is precluded by Claim 4.8.   □

Lemma 4.10. *For any parking function f, we have $\mu(\pi(f)) = f$.*

Proof. Let T be the arborescence formed by rooting the tree $\pi(f)$ at q. Fix a vertex u. From Claim 4.9 we have that μ will first remove all the edges in E_u. After the removal of these edges, u will dominate the remaining $f(u) + 1$ vertices in X_u and thus its out-degree will be that. Thus, $\mu(\pi(f))(u) = \text{out-degree}_D(u) - 1 = f(u)$.   □

Lemma 4.11. *For any spanning tree Z, we have $\pi(\mu(Z)) = Z$.*

Proof. Let $f := \mu(Z)$. Abuse notation and call the arborescence obtained by rooting Z at q, also Z. Let E' be the set of edges which form broken circuits with Z. Let D be the di-graph obtained by orienting the edges of $E \setminus E'$ with respect to Z. Given a subset of vertices X, let $Z[X]$ be the induced sub-forest of X.

The proof proceeds by induction on the stages of the algorithm computing π. We assume at stage i, the current tree of the algorithm, T, is a subtree of the tree Z. That is $T = Z[X_i]$.

We also assume that for every vertex in X_i, all its out-neighbors in D are also in X_i. These are vacuously true at stage 0. We now show that at stage $i + 1$ a vertex connects to tree T using an edge of Z and all its out-neighbors of D are in X_i.

Consider vertices in $V \setminus X_i$. At least one of these vertices v must have all its $f(v) + 1$ out-neighbors of D in X_i for otherwise we would get a cycle in D. Call this set of vertices S. Note that $Z[X_i \cup S]$ is connected as the vertices in X can only connect to Z using some edge of D. Let $u^* \in S$ be the vertex which is dominated by all other vertices in S in $Z[X_i \cup S]$. Let (u^*, w^*) be the edge in Z with $w^* \in X_i$. We claim that the algorithm which computes π also picks u^* in this stage and w^* is $M(u)$.

Note that $S \subseteq S_{i+1}$ since each vertex in S has at least $f(u) + 1$ neighbors in X_i. In fact, we show for every vertex $u \in S$, the corresponding w in X_i where $(u, w) \in Z$ is in fact $M(u)$. Call the set of these $f(u) + 1$ out-neighbors of u in X_i, Y_u. Observe that any edge of the form (u, v) with $v \notin Y_u$ must form a broken cycle with Z. This is because these edges are not directed towards v and cannot be directed towards u by the induction hypothesis. Since they form broken cycles with Z, these $|X_u| - |Y_u|$ vertices must be more powerful than w with respect to the arborescence Z restricted to vertices of X_i, that is T by the induction hypothesis. This follows from the definition of power. Moreover, w must be powerful than all other vertices of Y_u with respect to T since u dominates all these vertices. Thus w is the $(|X_u| - |Y_u| + 1)$th powerful vertex in X_u with respect to T, that is, $w = M(u)$.

We will be done if we show any vertex $u' \in S_{i+1} \setminus Q$ dominates some vertex in S with respect to Z. If this is the case, then the algorithm would choose the vertex which is dominated by all vertices in S_{i+1} and it has to be the vertex u^*. But this is true since u' has some out-neighbor of D in $V \setminus X_i$ – a path following argument shows we must reach a vertex $v \in S$ from u' using edges in D. In other words, u' dominates v with respect to Z.   □

In Figure 2 below, we give an example of the bijection on a simple 4-vertex 5-edge graph.

We end this section by using the relation between parking function and orientations to prove a property about the poset formed by parking functions. Recall given two parking functions f and g, we say that $f \prec g$ iff $f(v) \leq g(v)$ for all vertices $v \in V$. Also recall the parking function $f \wedge g$ defined as $(f \wedge g)(v) := \min(f(v), g(v))$. Finally, recall $\mathrm{dom}(f) := \{g$ a parking function $: g \prec f\}$.

Lemma 4.12. *Given a non-maximum parking function g, let F_g be the set of maximum parking functions that dominate g. Then, $g = \bigwedge_{f \in F_g} f$.*
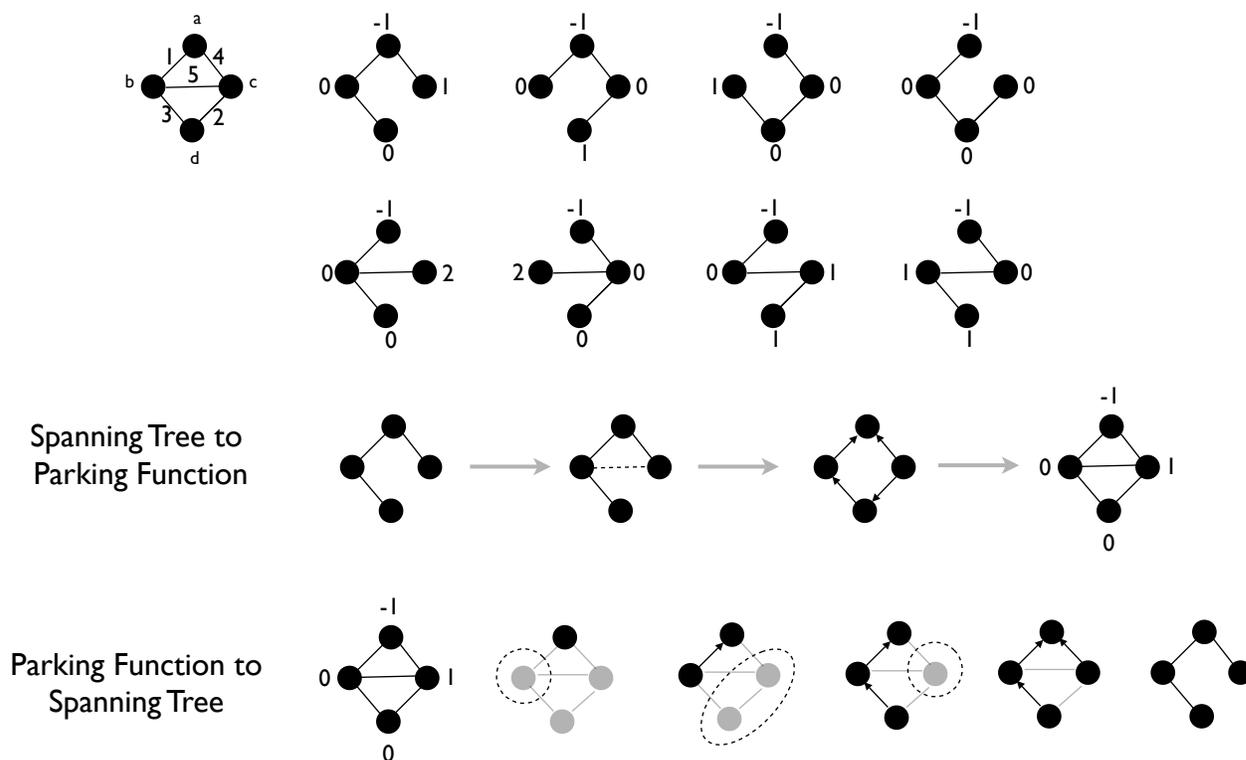
FIGURE 2. We illustrate our bijection on the graph on the top left corner. The numbers on the edges is the ordering σ. The graph has 8 spanning trees and they are shown beside the graph. The numbers on each spanning tree correspond to the 8 possible parking functions of the graph. It is instructive to note that the second row of spanning trees correspond to safe trees w.r.t to σ and correspond to maximum parking functions – note the sum of the numbers is 1 for the bottom 4 trees. We now take the first tree and show how it leads to the parking function via the function μ and then how vice-versa is obtained via the function π. μ first recognizes the edges which form broken circuits with T – the dotted edge is the only one in this case. Then it orients the tree edges towards the root a. It orients the edge (c, d) towards d because $c >_\sigma d$ w.r.t T. After the orientation, the parking function is found by subtracting 1 from the out-degrees. Going from the parking function to the tree, the dark vertices denote the set X_i at each step, while the dotted lines enclose the set S_i. When S_i has size more than 1 (when it contains vertices $\{c, d\}$), it finds $M(c) = b$ and $M(d) = b$ according to the power-order. It then chooses (d, b) since c dominates d in the tree with both (c, b) and (d, b) added.

Proof. It is clear that any non-maximum parking function g is dominated by the meet of all maximum parking functions which dominate it. The lemma claims that it is in fact exactly equal to the meet. To show this, it suffices to show that for any v, there exists a parking function f such that $f(v) = g(v)$, $f(u) \geq g(u)$ for all $u \in V$ and $f(w) > g(w)$ for exactly one

vertex $w \neq v$. Continuing this process, we get a maximum parking function which agrees with g on v. Since this is true for all v, we are done.

To prove the above, note that from the bijection between parking functions and spanning trees, we see that any parking function g uniquely corresponds to a *mixed* orientation of the edges of G, where a *mixed* orientation has some edges directed and others undirected. Moreover, the directed edges induce a DAG.

Consider the mixed orientation with respect to g. Let D be the DAG induced by the set of directed edges. Suppose there is an edge (u, w) which is not oriented where u and w are distinct from v; then in D if there is a directed path from u to w, orient the edge from u to w, else orient the edge from w to u. if there is no path either way, orient in any of the two ways. We still have a mixed orientation where the directed edges are acyclic. Moreover, this increases the value of parking function on exactly one of u or w, and keeps the value the same everywhere else.

Thus, the non-trivial case is when the only undirected edges are incident on v. Let (u, v) be such an edge. Note that we do not want to increase $g(v)$, that is, we want to orient (u, v) without increasing v's out-degree. In other words, we want to orient it from u to v. This is a problem if and only if there is a path from v to u in D. If so, consider the longest such a path $P = (v, w, \ldots, u)$, where w is the neighbor of v on P.

Note that apart from the edge (v, w), there is no path from v to w. If there were such a path, it can not use vertices from P, for w "dominates" (in the DAG) all vertices in P other than v. Thus, the path from v to w must use "new" vertices making P longer, and thus providing a contradiction.

Therefore, we can flip (v, w) to (w, v) and not create any cycles. That is, the operation to get f is to flip (v, w) to (w, v) and orient (v, u) from v to u. This only increases $g(w)$ but keeps everything else the same. □

5. Product Graphs and Q_n-Parking Functions

Given two graphs G_1 and G_2, there is a standard notion of the Cartesian product $G_1 \square G_2$ of the two graphs. Given a G_1-parking function and a G_2-parking function, we define below a $G_1 \square G_2$-parking function in a natural way that is symmetric in G_1 and G_2.

Definition 5.1. *Given $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the Cartesian product graph $G_1 \square G_2 = (V, E)$ is defined on the vertex set $V = V_1 \times V_2$, using the edge set $E = E' \cup E''$, where*

$E' = \{\{(u_1, v), (u_2, v)\} : \{u_1, u_2\} \in E_1 \text{ and } v \in V_2\}$, and $E'' = \{\{(u, v_1), (u, v_2)\} : \{v_1, v_2\} \in E_2 \text{ and } u \in V_1\}$.

It is easy to see from the definition that the number of vertices in $G_1 \square G_2$ is $|V_1||V_2|$, and that the number of edges is $|V(G_2)||E(G_1)| + |V(G_1)||E(G_2)|$.

5.1. **Parking functions on Product graphs.**

While it seems hard to characterize parking functions on $G_1 \square G_2$, by simply knowing those on G_1 and on G_2, the following result gives an explicit way to construct a particular family of maximum parking functions on product graphs.

Proposition 5.1. *Let* $f_1 \in \mathcal{P}(G_1; q_1)$ *and* $f_2 \in \mathcal{P}(G_2; q_2)$. *Then* $f_1 \square f_2 \in \mathcal{P}(G_1 \square G_2; (q_1, q_2))$, *where* $f_1 \square f_2(u, v) = f_1(u) + f_2(v) + 1$, *whenever* $u \in G_1$ *and* $v \in G_2$.

Further, if f_1 *and* f_2 *are maximum parking functions, then* $f_1 \square f_2$ *is a maximum parking function.*

Proof. Let $f = f_1 \square f_2$, $G = G_1 \square G_2$, and $q = (q_1, q_2)$. To show that $f \in \mathcal{P}(G, q)$, once again we make crucial use of Dhar's marking algorithm; in particular, we will make use of the order in which the vertices of G_1 (and G_2) are marked in verifying that $f_1 \in \mathcal{P}(G_1, q_1)$ (and $f_2 \in \mathcal{P}(G_2, q_2)$, respectively). Using these in turn, we order the vertices in $G_1 \square G_2$: let $(u', v') <_\square (u, v)$, if u' is marked before u in G_1, or if $u = u'$ and v' is marked before v in G_2. We now prove that f is a valid parking function, by showing that the vertices of $G_1 \square G_2$ can be marked, using Dhar, in precisely the order given by $<_\square$. We begin by noting that $f(q) = f(q_1, q_2) = f_1(q_1) + f_2(q_2) + 1 = -1$. Now consider the vertices of G inductively, using the order given by $<_\square$. By the time the vertex (u, v) is considered, observe that there are at least $f(u) + f(v) + 2$ neighbors of (u, v) that have already been marked, since they precede (u, v) in $<_\square$: indeed, at least $f(u)+1$ neighbors of the form (u', v) have been marked, (since the graph induced by fixing the second coordinate v is simply G_1), and similarly at least $f(v)+1$ neighbors of the form (u, v') have also been marked. Thus (u, v) can be assigned the value $f(u, v) = f(u) + f(v) + 1$, and hence f is a G-parking function.

Now, suppose that f_1 and f_2 are *maximum* G_1- and G_2-parking functions respectively. Then, we must show that f achieves the genus of $\mathbf{g}(G)$ minus one. This is easy to verify:

$$\begin{aligned} \|f_1 \square f_2\| &= |V(G_2)|\|f_1\| + |V(G_1)|\|f_2\| + |V(G_1)||V(G_2)| \\ &= |V(G_2)||E(G_1)| + |V(G_1)||E(G_2)| - |V(G_1)||V(G_2)| \\ &= |E(G_1 \square G_2)| - |V(G_1 \square G_2)| = \mathbf{g}(G_1 \square G_2) - 1. \end{aligned}$$

5.2. Q_n-Parking Functions. A quintessential product graph is the n-dimensional cube Q_n, obtained by taking the product of an edge Q_1 with itself n times. For the purpose of this section, for integer $n \geq 1$, we also view the n-cube Q_n as the graph on 2^n vertices, which may conveniently be labeled by the 2^n binary vectors of length n, and with edges between vertices whose Hamming distance is one. We are interested in understanding the parking functions on Q_n with respect to the vertex $q = (0, 0, \ldots, 0)$.

Definition 5.2. *For $n = 1$, the unique parking function $f = f^1$ on Q_1 is canonical. For $n > 1$, the parking function $f^n = \underbrace{f \square f \square \cdots \square f}_{n \text{ times}}$, obtained using the product graph construction, is defined as the* canonical *Q_n-parking function. Further, if a Q_n-parking function g is such that there exists a bijection $\phi : V(Q_n) \to V(Q_n)$ such that $f(v) = g(\phi(v))$ for all $v \in V(Q_n)$, then we say that g is* semi-canonical.

Note that by Proposition 5.1, the canonical Q_n-parking function is a maximum parking function, and hence the semi-canonical one is also a maximum.

Example 5.1. Not *all maximal Q_n-parking functions are semi-canonical. For instance, consider the Q_3-parking function f such that $f(000) = -1, f(001) = 1, f(010) = 0, f(100) = 2, f(011) = 0, f(101) = 0, f(110) = 0, f(111) = 2$. It is easy to verify that f is a Q_3-parking function, but f is not semi-canonical. Since, $\|f\| = 4 = |E| - |V|$, f is maximum (and thus maximal).*

Recall the partial order on parking functions, and the notion of $\mathrm{dom}(f) = \{g \in \mathcal{P}(G, q) : g \prec f\}$, for a parking function $f \in \mathcal{P}(G, q)$.

Theorem 5.2. *Let f^n denote the canonical Q_n-parking function. Then $f^n(v) = \mathrm{wgt}(v) - 1$, where $\mathrm{wgt}(v)$ is the Hamming weight (the number of 1's in the binary representation) of the vertex $v \in V(Q_n)$. Consequently, if f is semi-canonical, then*

$$|\mathrm{dom}(f)| = \prod_{k=2}^{n} k^{\binom{n}{k}}.$$

Proof. The proof is by an easy induction on $n \in \mathbb{Z}^+$. The base case is clear, since f^1 takes the values $-1, 0$. For the induction step, for $n \geq 1$, write $f^{n+1} = f^n \square f^1$, which by Proposition 5.1 implies that, for $u \in V(Q_n)$ and $v \in V(Q_1) = \{0, 1\}$,

$$f^{n+1}(u, v) = f^n(u) + f^1(v) + 1.$$

Thus for $0 \le k \le n$, the value k can be obtained in f^{n+1} either by taking a k from f^n and adding a zero to it (through, $f^1(0) + 1 = -1 + 1$), or by taking a $k - 1$ from f^n and adding a one to it (through, $f^1(1) = 0 + 1$). By the induction hypothesis, the number of k's is

$$\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k},$$

completing the induction step. Also note that, by definition, a semi-canonical parking function also has the same distribution of integers.

To complete the proof of the theorem, recall by Proposition 2.2, that for any $v \in V(Q_n) \backslash \{q\}$, if $f(v) = k$ for $k \in \mathbb{Z}_{\ge 1}$, then for $g \prec f$, we may have $g(v) = 0, 1, \ldots k$; this implies that there are $k + 1$ possible values for such a g with respect to v. Since $0 \le k \le n$ and each value in the range of f is independent with respect to obtaining a dominated function g, we have that

$$|\text{dom}(f)| = \prod_{k=2}^{n} k^{\binom{n}{k}}.$$

∎

Remark 5.1. *Note that it can directly be seen that f is a Q_n-parking function, when f is defined as $f(v) = \text{wgt}(v) - 1$. Using Dhar's algorithm, starting with $q = (0, 0, \ldots, 0)$, we may proceed by marking vertices level by level (in the Boolean lattice): Every vertex v in level k has precisely k down-neighbors in level $k - 1$, and they have all been marked, so v can be marked and given value $k - 1$. Thus there are precisely $\binom{n}{k}$ vertices which obtain the value $k - 1$, in such a canonical parking function on Q_n.*

Example 5.2. *Note that* not *every semi-canonical parking function is canonical – in satisfying the level by level property described in Remark 5.1. For instance, consider the Q_3-parking function f such that $f(000) = -1$, $f(010) = f(100) = f(101) = 0$, $f(001) = f(011) = f(111) = 1$, and $f(110) = 2$. It is easy to check that f is a Q_3-parking function, semi-canonical, but not canonical – since, for example, the value 2 is adjacent to a 0.*

We hope the above remarks and examples indicate the difficulty in understanding the *maximum* parking functions on even a highly structured, symmetric graph such as the *n*-cube. As far as we know, the number of maximum parking functions of Q_n is known only for $n \le 4$. For $n = 2, 3$, and 4, this number is 3, 133, and 3040575, respectively.

6. Diffuse states and acyclic orientations

As mentioned in the introduction, in the context of chip-firing, the following notion was introduced in [23].

Definition 6.1. *Given a connected graph G, a function $s : G \to \mathbb{Z}^+$ is called a diffuse state if for every induced subgraph $G[A] = (A, E(A)) \subseteq G$, there exists some $u \in A$ such that $\deg(u)|_A \leq s(u)$. Further, let $\|s\| := \sum_v s(v)$.*

Note the (complementary) similarity with the definition of a parking function, by observing that $\deg(u)|_A \leq s(u)$ is equivalent to $\deg(u) - s(u) \leq \deg(u)|_{A^c}$. However, we have no special vertex such as q. For chip-firing purposes, $s(v)$ may be thought of as the number of chips on v, thus $\|s\|$ denotes the total number of chips in the graph.

First consider the following algorithm which constructs an acyclic orientation, by using a given diffuse state s with $|E(G)|$ chips. We will make use of this in proving the necessary bijection of this section.

Step 1. Since the state s is diffuse, we can find a vertex $v \in V(G)$ such that $\deg(v)|_G = \deg(v) \leq s(v)$.

Step 2. Orient all of the edges incident to v outward.

Step 3. Delete v (and all of its incident edges).

Step 4. The new graph is diffuse since it is a subgraph of G, so we can repeat Steps 1-3 until all edges of the graph are oriented.

Note that this process gives an acyclic orientation since we cannot orient edges into a vertex which has out edges since this vertex has been deleted from the graph.

Lemma 6.1. *If s is a diffuse state on graph G with $\|s\| = |E(G)|$, then there is a vertex v with $s(v) = 0$. For every diffuse state s, the set $\{v : s(v) = 0\}$ is an independent set in G.*

Proof. The above algorithm which repeatedly removes vertices, removes at least as many chips as the edges at each step. Before the last vertex, all edges (hence all $|E(G)|$ chips) must have been removed, which means that the last vertex can has zero chips. For the second part, if $s(u) = s(v) = 0$, and $u, v \in E(G)$ then the set $A = \{u, v\}$ violates the diffuse property. ∎

With a similar proof, it can also be shown that there exists a vertex w such that $s(w) = \deg(w)$, under the hypothesis of the above lemma.

Theorem 6.2. *There is a bijection between the set* D(*G*) *of diffuse states with* $|E(G)|$ *chips and the set* AO(*G*) *of acyclic orientations of a connected graph G.*

Proof. Let $E(G) = m$. The proof is based on two injections between the sets, going in each direction. First, given an acyclic orientation $O \in \text{AO}(G)$, define the nonnegative function $s = s_O$ by letting $s(v)$ be the out-degree of v in the orientation O. Clearly, the mapping is one-one, since the out-degree sequence uniquely determines an acyclic orientation; also $\|s\| = m$, since the sum of out-degrees equals the number of edges. To see that s is diffuse, simply observe that, each induced subgraph $G[A] \subseteq G$ has a (local) source $y \in A$ when restricted to the acyclic orientation induced on A; such a source y satisfies $s(y) \geq \deg(y)|_A$, since the out-degree of y is at least the degree $\deg(y)|_A$.

For an injection in the other direction, we make use Lemma 6.1. Given a diffuse state s with m chips, we construct an acyclic orientation, by constructing a DAG: Lemma 6.1 guarantees the existence of sink(s); so we construct the orientation, by (i) repeatedly removing the current set of sinks, and (ii) subtracting a chip from each in-neighbor of a removed sink. It is easy to see that the updated function s at each step is still a diffuse state on the remaining graph. Note that this construction is one-one, since the DAG representation would come out differently, if s,s' are different diffuse states – this can be seen by considering the first time the current sets of sinks differ, when we start with s versus s'; since the underlying graph is the same, there must be such a time whenever $s \neq s'$. ■

Remark 6.1. *Thanks again to an anonymous referee of* [6]*, we learnt the following: that diffuse states are also in 1-1 correspondence with the critical configurations of the chip-firing game of Björner-Lovász-Shor* [9]*, and that the first part of the above proof appears as Theorem 3.3, part (b) in* [9]*. In addition, it is easy enough to go between the above theorem and Theorem 3.1: simply add a new vertex q adjacent to every vertex in G and obtain a new graph* $\hat{G}$*; then* $\hat{G}$*-parking functions (with respect to q, say) correspond to diffuse states in G, and acyclic orientations with a unique source at q in* $\hat{G}$ *correspond to acyclic orientations of G.*

7. Concluding Remarks

In addition to the questions mentioned in the previous sections, several challenging problems remain open. Given an arbitrary graph *G*, it is a classical open problem in the topic of Markov chain Monte Carlo (MCMC) algorithms [24], to efficiently generate an acyclic orientation *uniformly at random* from the set of all such orientations. Due to the observations above, a

closely related problem would be to generate at random an acyclic orientation with a uniquely identified sink (or source), or equivalently, to sample uniformly from the set of *safe* spanning trees of G. The MCMC technique suggests the following natural approach to this problem: it is well known (see for example, [16, 17]) that the so-called bases exchange walk provides an efficient way to sample uniformly from the set of *all* spanning trees of a given graph G. However it remains to be seen whether (and how) restricting such a random walk to the set of safe trees affects the *mixing time* of the walk – the time by which the walk converges to its steady state distribution, uniform on the set of safe trees. Given that the exact enumeration of the number of safe trees of Q_n is also open, an interesting first step might be to analyze such a walk on the trees of Q_n.

Independent of the above approach, other ways of providing asymptotically accurate estimates, for large n, of the number (or even the logarithm of the number) of maximum parking functions on Q_n, remains interesting and presumably a challenging exercise.

References

[1] P. Bak, C. Tang, and K. Wiesenfeld, *Self-organized criticality: an explanation of the $1/f$ noise*, Phys. Rev. Lett. **59** (1987), 381–384.

[2] M. Baker and S. Norine, *Riemann-Roch and Abel-Jacobi Theory on a Finite Graph*, Advances in Mathematics **215** (2007), pp. 766–788.

[3] M. Baker, Personal Communication.

[4] R. Bari, *A combinatorial approach to graphical polynomials and spanning subgraphs*, Topics in graph theory (New York, 1977), pp. 21–29, Ann. New York Acad. Sci., 328, New York Acad. Sci., New York, 1979.

[5] R. Bari, Chromatic Polynomials and the internal and external activities of Tutte, *Graph theory and related topics*, pp. 41–52, Academic Press (1979), New York.

[6] B. Benson and P. Tetali, *Parking Functions and Acyclic Orientations of Graphs*, math arXiv ..., 2008.

[7] N. L. Biggs, *Chip-Firing and the Critical Group of a Graph*, J. of Algebraic Comb. **9** (1999), pp. 25–45.

[8] N.L. Biggs, *The Tutte polynomial as a Growth function*, J. of Algebraic Comb. **10** (1999), pp. 115–133.

[9] A. Björner, L. Lovász and P. Shor, *Chip-firing games on graphs*, Euro. J. Combin. **12** (1991), 283–291.

[10] A. Blass and B. Sagan, *Bijective proofs of two broken circuit theorems*, J. Graph Theory **10** (1986), pp. 15–21.

[11] D. Chebikin and P. Pylyavskyy, *A family of bijections between G-parking functions and spanning trees*, Jour. Combin. Theory (Series A) **110** (2005), pp. 31–41.

[12] R. Cori, Y. Le Borgne, *The sand-pile model and Tutte polynomials*, Adv. Appl. Math. **30** (2003), pp. 44–52.

[13] R. Cori and D. Rossin, *On the sandpile group of dual graphs*, Euro. J. Combin. **21** (2000), pp. 447–559.

[14] R. Cori, D. Rossin, B. Salvy, *Polynomial ideals for sandpiles and their Grobner bases*, Theor. Comp. Sci. **276** (2002), pp. 1–15.

[15] D. Dhar, *Self-organised critical state of the sandpile automaton models*, Physical Review Letters **64** (1990), pp. 1613–1616.

[16] T. Feder and M. Mihail, *Balanced Matroids*, Proc. of the 24th Annual ACM Symp. on Theory of Computing (1992), 26–38.

[17] J. Fehrenbach and L. Rüschendorf, *Analysis of Markov chain algorithms on spanning trees, rooted forests, and connected subgraphs*, Appl. Math. (Warsaw) **32** (2005), 341–365.

[18] A. Gabrielov, *Abelian Avalanches and Tutte Polynomials*, Physica A **195** (1993), 253–274.

[19] D. D. Gebhard and B. E. Sagan, *Sinks in Acyclic Orientations of Graphs*, Jour. of Combin. Th. (Series B) **80** (1999), pp. 130–146.

[20] E. Gioan and M.L. Vergnas, *Activity preserving bijections between spanning trees and orientations in graphs*, Discrete Math. **298** (2005), pp. 169–188.

[21] C. Godsil and G. Royle, Algebraic Graph Theory, *Springer-Verlag* GTM Vol. 207 (2001), New York.

[22] C. Greene and T. Zaslavsky, *On the Interpretation of Whitney Numbers through Arrangements of Hyperplanes, Zonotopes, Non-Radon Partitions, and Orientations*, Trans. Amer. Math. Soc. **280** (1983), pp. 97–126.

[23] J. Jeffs and S. Seager, *The chip firing game on n-cycles*, Graphs Combin., **11** (1995), pp. 59–67.

[24] M. Jerrum and A. Sinclair, *The Markov chain Monte Carlo method: an approach to approximate counting and integration.* In D. Hochbaum (Ed.), Approximation Algorithms for NP-hard problems, PWS Publishing (1996), pp. 482–520.

[25] R. Kenyon and P. Winkler, *Branched Polymers*, Amer. Math. Monthly, to appear.

[26] C. M. Lopez, *Chip Firing and the Tutte Polynomial*, Annals of Combinatorics **1** (1997), pp. 253–259.

[27] J. Plautz and R. Calderer, *G-Parking Functions and the Tutte Polynomial*, Preprint.

[28] A. Postnikov and B. Shapiro, *Trees, parking functions, syzygies, and deformations of monomial ideals*, Transactions of the American Mathematical Society 356 (2004), pp. 3109–3142.

[29] R. Stanley, Enumerative Combinatorics, Vol. II, *Cambridge Studies in Advanced Mathematics* **Vol. 62**, Cambridge University Press (1999), Cambridge.

[30] W.T. Tutte, *A Contribution to the Theory of Chromatic Polynomials*, Canad. J. Math. **6** (1954), pp. 80–91.

[31] D.J.A. Welsh, Complexity: Knots, Colourings, and Counting, *London Math Soc. Lecture Note Series* **Vol. 186**, Cambridge University Press (1993), Cambridge.

[32] H. Whitney, *A Logical Expansion in Mathematics*, Bull. Amer. Math. Soc. **38** (1932), pp. 572–579.

B. Benson, Department of Mathematics, University of Illinois, Urbana-Champaign, Illinois

E-mail address: benson9@illinois.edu

D. Chakrabarty, Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G, Canada

E-mail address: deepc@math.uwaterloo.ca

P. Tetali, School of Mathematics and School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332-0160.

E-mail address: tetali@math.gatech.edu